1.中间件

拦截对服务器端的请求，也可以作出响应

express下中间件分为应用级中间件、路由级中间件、内置中间件、第三方中间件、错误处理中间件

(1)应用级中间件

也称为自定义中间件，本质上就是一个函数

app.use(URL, 回调函数) 拦截特定的请求

app.use(回调函数) 拦截所有的请求

练习：添加到购物车路由(get /shopping)，传递商品的价格price；添加中间件对传递的价格打9折；最后在路由中显示以下效果

商品的价格：900

http://127.0.0.1:8080/shopping?price=1000

(2)路由级中间件

app.use(拦截的URL, 路由器)

(3)内置中间件

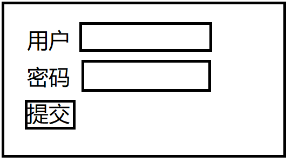
托管静态资源

客户端请求静态资源(html,css,js,图像...)，不需要再创建路由，让客户端自动的到指定的目录下查找。

app.use( express.static('要托管的目录') );

练习：将静态资源托管到files目录

练习：编写文件04\_three.js，创建WEB服务器，托管静态资源到public目录，目录中含有登录的网页login.html，点击提交向服务器发请求(post /mylogin)，获取传递的数据，响应‘登录成功，欢迎：xx’



(4)第三方中间件

第三方中间件以第三方模块的形式出现，需要先下载安装

|  |
| --- |
| //1.引入中间件模块  const bodyParser = require('body-parser'); |
| //2.使用中间件，将所有post请求的数据解析为对象  //把对象放入到了请求对象下 req.body  app.use( bodyParser.urlencoded({  extended: false //false不使用第三方的查询字符串模块，就会使用核心模块querystring  }) ) |
| //3.在路由中获取数据，格式为对象  req.body |

2.mysql模块

Node.js下，专门用于操作mysql数据库的第三方模块

npm install mysql 下载安装

mysql命令

mysql -uroot

mysql.exe -h127.0.0.1 -P3306 -uroot -p

mysql -uroot<拖拽脚本

insert into emp values(...);

delete from emp where eid=5;

update emp set sex=1,... where eid=7;

select \* from emp;

select \* from emp where ename='lucy' or 1 ;

SQL注入：在让用于提供的值中，拼接了其它的SQL命令。

占位符（?）: 先对数据进行过滤，过滤完以后再替换占位符

mysql.createConnection() 创建连接对象

connect() 测试连接

query(SQL命令, 要过滤的数据, 回调函数)

课后任务

(1)复习今天内容，整理思维导图

(2)练习：创建web服务器，托管静态资源到public目录，包含部门添加的网页add.html，点击提交，向服务器发请求(post /add)，获取传递的数据，将数据插入到tedu下dept表中，响应'部门添加成功'

